A simplified MOLP algorithm

Strijbosch, L.W.G.; van Doorne, A.G.M.; Selen, W.J.

Publication date:
1990

UNIVERSITY
UNIVERSITEIT
BRABANT

POSTBOX 90153
5000 LE TILBURG
THE NETHERLANDS

[redacted]

DEPARTMENT OF ECONOMICS

RESEARCH MEMORANDUM

[redacted]

A SIMPLIFIED MOLP ALGORITHM:
THE MOLP-S PROCEDURE

Leo W.G. Strijbosch,
Arno G.M. van Doorne, Willem J. Selen

FEW 460

Oktober 1990

A simplified MOLP algorithm: The MOLP-S procedure.

Leo W.G. Strijbosch†‡,

Arno G.M. Van Doorne*
Faculty of Economics
Tilburg University
5000 LE Tilburg
The Netherlands

Willem J. Selen §
Boston University - Brussels
Laarbeeklaan 121
1090 Brussels
Belgium

† Leo Strijbosch is a staff member at the Faculty of Economics at Tilburg University, responsible for information system support. He holds an engineering degree from the Technical University of Eindhoven and a Ph.D in medical statistics from the University of Limburg. Dr. Strijbosch has previously published in Biometrics, Communications in Statistics - Simulation and Computation, International Journal of Biomedical Computing, Statistics in Medicine, among others.

* Arno Van Doorne is a Drs-student at Tilburg University, working towards a specialization in business econometrics.

§ Willem Selen is Director of Boston University - Brussels and Professor of Management Science. He holds a commercial engineering degree from the Limburg Business School and a Ph.D in management science from the University of South Carolina. Dr. Selen has previously published in the Journal of Information Science, the European Journal of Operational Research, the Journal of the Operational Research Society, Applied Mathematical Modelling, among others.

‡ To whom correspondence should be addressed.

A simplified MOLP algorithm: The MOLP-S procedure.

Leo W.G. Strijbosch,
Arno G.M. Van Doorne
Faculty of Economics
Tilburg University
5000 LE Tilburg
The Netherlands

Willem J. Selen
Boston University - Brussels
Laarbeeklaan 121
1090 Brussels
Belgium

Scope and Purpose - The pupose of this article is to describe a simplified algorithm which can be used to solve multiple objective linear programming problems. The method is based on a straightforward extension of the simplex-method, and may prove to be computationally more efficient as compared to the well-known ADBASE algorithm.

Abstract - A number of MOLP-algorithms have been developed to establish the set of non-dominated solutions, using a number of different approaches and theorems that may be non-trivial to the non-expert user. This article presents a simplified MOLP-algorithm (MOLP-S), based on a straightforward extension of the simplex-method of linear programming, to trace out the set of non-dominated solutions. The proposed methodology exhibits computational characteristics that may render the method more efficient as compared to other algorithms currently in use. The proposed method is tested on a number of problems from the literature which exhibit varying degree of complexity.

A simplified MOLP algorithm: The MOLP-S procedure.

1. Introduction

Management often finds itself confronted with multiple and conflicting objectives in modeling and establishing a structured problem solving approach to many business problems. A traditional example involves the attainment of high quality of service under set budget constraints, while minimizing total cost. When the objectives and constraints can be assumed as, or approximated by, linear functions the problem is said to be a multiple objective linear programming problem (MOLP). The MOLP-problem is mathematically defined* as:

Max Cx

Subject to $x \in S$

where C = k*n matrix of objective function coefficients,

A = m*n matrix of constraint function coefficients,

$S = \{x \in R^n \mid x \geqq 0,\ Ax=b,\ b \in R^n\}$.

The solution of an MOLP-problem consists of the enumeration of all non-dominated (or *efficient*) solutions.

Definition 1.

An extreme point $x \in S$ is defined to be efficient if and only if there does not exist an $\tilde{x} \in S$ such that $C\tilde{x} \geq Cx$.

Current MOLP-algorithms consist of three major steps:

Step 1: Determination of an initial feasible point, similarly to linear programming.

Step 2: Determination of an initial efficient extreme point.

Step 3: Determination of the complete set of efficient points.

---

* We use the following convention for vector inequalities: $x \geqq 0$ if $x_j \geqq 0$, $j=1,\ldots,n$; $x \geq 0$ if $x_j \geqq 0$, $j=1,\ldots,n$, $x \neq 0$.

Step 2 consists of pivoting from the initial feasible extreme point of step 1 to an efficient extreme point, if one exists. Step 3 consists of generating all efficient extreme points, using the initial efficient extreme point as the starting solution, as all efficient points form a connected set. Hwang and Masud [1] describe different approaches for tracing out the set of efficient points, such as parametric programming, the adjacent efficient basis-approach and adjacent efficient extreme point-approach. A number of algorithms were developed to determine the complete set of efficient extreme points, varying in complexity and computational efficiency. This study presents an algorithm which is logically transparant to the non-expert user who is familiar with the well-known Simplex method of linear programming. Furthermore, a number of test problems from the literature are used to demonstrate the computational efficiency of the proposed algorithm.

2. Problem definition

Some well-known algorithms for solving MOLP-problems are:

a) ADEX-procedure [2],

b) ADBASE-procedure [2],

c) Multicriteria Simplex Method [3],

For details on the multicriteria simplex method, we refer to Zeleny [3]. We find that this method is not apparantly transparant to the non-expert user and involves numerous procedures to check on the efficiency of a particular basis. Moreover, limited experience with this method shows that, especially with degenerated problems, CPU-time is large as compared to other procedures. The ADEX procedure is based on an adjacent efficient <u>extreme point</u> approach for generating the entire set of efficient solutions. This method determines all possible directions emerging from an existing efficient point leading to the other extreme points, and tests each direction for efficiency. The ADBASE method is based on an adjacent efficient <u>basis</u> approach. This method involves pivoting out of all efficient bases. If an extreme point is efficient and not degenerated, the corresponding basis is efficient too. A degenerated efficient extreme point, however, has at least one efficient basis corresponding with it, while other could be dominated. Evans and Steuer [2] pointed out that the

ADBASE procedure proved computationally more efficient as compared to ADEX, making ADBASE the more promising procedure. In this research we present a modified version of the ADBASE procedure, called MOLP-S, and compare both methods in terms of computational efficiency on a number of test problems from the literature.

The following definition can be used to verify whether or not a basis is efficient (see [2]):

Definition 2: A basis is efficient if no $v \in R^k$ exists for which $e^T v > 0$ in

$$\text{Max } e^T v$$
$$\text{subject to } Wy + Iv = 0$$
$$y, v \geq 0$$

where $e \in R^k$, a vector of ones,

$I \in R^{k*k}$, the identity matrix,

$W \in R^{k*d}$, the reduced cost matrix

($d = n - m$, the number of nonbasic variables),

$y \in R^d$, $v \in R^k$, vectors with dummy variables.

This definition provides a computational test to verify wether the system $Wy \leq 0$ is consistent (in which case the basis is dominated). We only need this test for establishing the first efficient basis. Out of each efficient basis, all bases that could be reached within one Simplex-iteration but are not in a list of efficient bases already found, are determined and tested for efficiency with help of the following theorem (c.f. [2]):

Theorem 1: Given an efficient basis B, x_j a nonbasic variable, pivoting x_j into the basis would result in an efficient basis B' if the following condition is satisfied:

No $v \in R^k$ exists for which $e^T v > 0$ in

$$\text{Max } e^T v$$
$$\text{subject to } Wy - w_j r + Iv = 0$$
$$y, v \geq 0$$
$$r \geq 0$$

where w_j, column in reduced cost matrix W corresponding to x_j, $w_j \in R^k$,

r, dummy variable, $r \in R$.

The condition in theorem 1 is sufficient but not necessary [4]. If B is efficient and B' is efficient according to theorem 1, let us call (B,B') an efficient pair. The graph G=(V,E), where V is the set of efficient bases and E the set of efficient pairs, is connected [5]. Thus, starting from an arbitrary efficient basis, all remaining efficient bases can be found using the computational test of theorem 1 by checking all adjacent bases from each efficient basis. Note that the test is only needed once for all adjacent bases that can be reached with the same pivotcolumn (in case of degeneracy). However, a basis must be checked from each adjacent (efficient) basis until it turns out to be efficient. The fact that each efficient point has at least one efficient basis guarantees that we find all efficient points. The search proces of efficient bases can be implemented in different ways, see section 3.

A typical computational procedure for mapping out the complete set of efficient bases consists of three steps:
Step 1 - Determination of a basic feasible solution (BFS).
Step 2 - Determination of an initial efficient basis.
Step 3 - Search for subsequent efficient bases.
We elaborate on step 3 where computational efficiency is of particular importance, especially for large problems. Inherent to the implementation of step 3 are two important computational issues: 1) the significant bookkeeping task required to avoid revisiting a previously computed efficient basis; 2) the accumulation of inaccuracies resulting from the long sequence of Simplex-operations necessary in the search process for all efficient bases.
The first problem was solved by denoting a BFS uniquely as the set of indices of corresponding nonbasic variables and building "linked lists" of these sets. The main advantage of using linked lists is that they do not demand a priori fixed memory requirements. In order to avoid the searching of one large linked list to discover eventually that a given BFS had not been registered, a partition of the complete set of BFS's is made so that a certain search action only requires inspection of one (small) linked list. A technique, referred to as "hashing", can be used to accomplish

this by taking the sum of the indices of the nonbasic variables, or this sum modulo 200, for example. Figure 1 illustrates the linked list principle. The index Si200 is determined as the sum of the indices of the nonbasic variables modulo 200. A pointer associated with this index (Root[Si200]) either refers to nothing (NIL) or points to a record containing a set of indices of nonbasic variables (set of nbv) and a pointer. The last record in this range always has a pointer referring to NIL. Thus searching and updating can be performed very fast.

Figure 1. *Illustration of the linked list principle for searching and updating a list of BFS's.*

Si200	Root[Si200]	Records
0	NIL	
.	.	...
.	.	...
j	pointer	→ [set of nbv, pointer] →...→ [set of nbv, NIL]
.	.	...
.	.	...
199	pointer	→ [set of nbv, NIL]

The second problem was resolved by ensuring that a reinversion, required in the Simplex method for mathematical programming, was automatically performed after a predetermined number of Simplex-iterations. For that purpose we only need the original tableau and the actual set of nonbasic variables. There is no need to store an inverse basis matrix for each unprocessed efficient basis (c.f. [5]).

The next section describes and compares two approaches which are very much alike, both using the linked list and reinversion technique, but differ in the way subsequent efficient bases are established. One algorithm is essentially ADBASE [2], while the other is a new variant that uses a procedure which determines a path of efficient BFS's along which the efficient points are established. This latter procedure is denoted as MOLP-S.

3. Two algorithms for generating the set of efficient bases

Both the MOLP-S and the ADBASE procedure determine, out of each efficient basis, all adjacent bases which have not been registered in a set S as "efficient bases". For all these bases, it is verified using theorem 1 wether or not they form an efficient pair with the actual basis. Only efficient adjacent bases are subsequently stored in the set S. At this point the ADBASE and MOLP-S variants proceed in a different way.

The ADBASE variant stores the efficient adjacent bases cumulatively in a list of unscanned efficient bases and pivots to the last one, which is subsequently removed from the list. The change of one efficient basis to the next will sometimes involve one Simplex-iteration but other times more than one (in particular when no new adjacent efficient basis can be reached). The process continues until the list of unscanned efficient bases is empty. ADBASE sometimes requires a number of Simplex-iterations just to locate a known efficient basis in order to continue its testing for more adjacent efficient bases.

The MOLP-S variant stores the efficient adjacent bases (all but one, which will be used for the forward step) as being unexplored at a place depending on the length of the path which is traced. The path is walked up (a forward step) when at least one new adjacent efficient basis is discovered, and walked down (a backward step) when such a basis does not exist. A forward step increases the length of the path with one, a backward step decreases it with one. The pivot which defines a forward step is stored in an array P so that a backward step can be performed immediately. The path being walked down will never be walked up again. Stepping backward continues until a basis is obtained for which at least one unexplored efficient adjacent basis exists; stepping forward continues as long as at least one new adjacent efficient basis can be reached. The MOLP-S procedure terminates as soon as it has returned to the initial efficient basis (path length is zero) and no new adjacent efficient basis can be located. These two variations for mapping out the set of efficient extreme points are technically described in detail in the Appendix for the interested reader.

Note that the MOLP-S algorithm traces a path through the connected set of efficient BFS's, going forward when an adjacent efficient BFS exists which has not yet been encountered, and going backward in absence of such a BFS. The ADBASE algorithm, on the contrary cumulatively stores all new adjacent efficient BFS's in one list and subsequently pivots to the last one. This may require more than one pivotstep. Of course it would be possible to search the list of unscanned BFS's for the one which is reachable within the least number of steps, but this process would probably use more computation time as compared to automatically pivoting to the last basis in the set of all new adjacent efficient BFS's. Consider a MOLP-problem solved by the two algorithms in exactly the same number of Simplex-iterations. A difference in CPU-time would then be accounted for by the time needed in ADBASE to determine the necessary pivots, as the MOLP-S algorithm does not have to determine these pivots (see Appendix). Two important computational issues will be looked at in greater detail in comparing the two variants. These are:

(a) Will ADBASE generally need more Simplex-iterations than MOLP-S?

(b) If ADBASE needs equal or even less Simplex-iterations than MOLP-S, what will be the influence of the extra required CPU-time to determine the pivots which consequently lead to the change of one basis to another?

A full comparison of the computational performance of both methods is beyond the scope of this research effort. However, a number of test-problems reported in the literature and exhibiting varying degrees of complexity, were run with both the ADBASE and the MOLP-S variant and are described next.

4. Computational experience

A number of test problems from the literature were used to verify the two variants, and to indicate some of the computational requirements. The test problems, together with some of their characteristics, are listed in Table 1. Note that example 17 correctly exhibits 29 efficient extreme points, instead of 70 efficient points reported on by Zeleny [3]. The extra efficient points were probably found due to numerical deficiencies. The same number of 29 efficient points were found earlier by Isermann [6].

Table 1. *Analyzing test problems from the literature.*

Test problem	Problem characteristics V	C	O	N	B	Computational characteristics T_1	I_1	T_2	I_2
1 [7] p. 313 (D)	3	4	2	4	6	9	13	9	9
2 [7] p. 316 (U)	2	2	2	2	2	5	3	5	2
3 [1] p. 258	2	2	2	2	2	6	4	6	3
4 [1] p. 271	4	3	3	4	4	11	7	11	4
5 [6] p. 243	3	4	3	6	6	11	12	10	7
6 [8] p. 128 (TP)	9	6	3	7	7	15	21	13	15
7 [9] p. 93	7	7	4	25	25	129	50	146	61
8 [10] p. 1096 (D)	3	5	3	6	8	12	15	13	13
9 [11] p. 358 (D)	2	3	2	1	1	8	2	5	2
10 [12] p. 67 (SD)	10	9	3	14	26	192	111	191	102
11 [13] p. 93	4	2	2	6	6	10	11	9	8
12 [14] p. 244 (D)	5	5	3	11	11	26	22	27	20
13 [14] p. 258 (D)	8	8	4	6	8	15	20	14	15
14 [14] p. 267	6	6	4	12	12	46	23	44	15
15 [3] p. 43 (SD)	7	4	3	6	17	48	33	51	32
16 [3] p. 115 (D)	8	8	3	3	3	16	15	16	13
17 [3] p. 117 (D)	8	8	5	29	29	225	67	231	65
18 [3] p. 141 (D)	3	3	3	5	7	12	15	12	13
19 [15] p. 228	2	3	2	3	3	6	6	6	4
20 [15] p. 230	2	6	2	3	3	7	8	5	6
21 [15] p. 233	2	3	2	1	1	5	2	4	2
22 [15] p. 265 (D,TP)	20	9	3	23	29	221	78	219	63
23 [15] p. 270 (D,TP)	8	6	3	4	4	9	16	8	15

D : degeneracy
SD: strong degeneracy
U : >1 objective functions unbounded
TP: transportation problem

V: # variables
C: # constraints
O: # objective functions
N: # efficient points
B: # efficient bases

T_i: cpu-time (0.01 sec.) of step 3 (i=1: MOLP-S; i=2: ADBASE)

I_i: # pivot-operations (i=1: MOLP-S; i=2: ADBASE)

Table 1 also reports some computational characteristics relating to each test problem, using a VAX-station 3100 (model 30). The programs were written in PASCAL. The results do not clearly indicate a preference for either algorithm. Therefore, the larger problems 6,7,10,13,14,15,16,17,22 and 23 were transformed to yield more efficient bases by using two objective functions, the first being the opposite of the second so that in fact all extreme points are generated. The results are presented in Table 2.

Table 2. *Some extended test problems of Table 1 with two opposite objective functions.*

Test problem	Problem characteristics V	C	O	N	B	Computational characteristics T_1	I_1	T_2	I_2
6 [8] p. 128 (TP)	9	6	2	18	18	18	43	17	33
7 [9] p. 93	7	7	2	118	118	178	236	222	285
10 [12] p. 67 (SD)	10	9	2	354	1294	3014	2597	3537	2729
13 [14] p. 258 (D)	8	8	2	8	12	16	28	13	23
14 [14] p. 267	6	6	2	32	32	39	63	43	60
15 [3] p. 43 (SD)	7	4	2	14	57	62	113	65	104
16 [3] p. 115 (D)	8	8	2	131	145	256	298	325	356
17 [3] p. 117 (D)	8	8	2	189	211	373	431	484	517
22 [15] p. 265 (D,TP)	20	9	2	885	2724	7754	5467	8417	5147
23 [15] p. 270 (D,TP)	8	6	2	12	12	11	32	11	26

Now computational differences become more apparant. The MOLP-S variant uses significantly less CPU-time than the ADBASE variant for problems 7,10,16,17 and 22, which turn out to be the largest problems. The savings are, respectively, 20,15,21,23 and 8%. Although sometimes (cf. problem 22) MOLP-S involves more Simplex-iterations, it remains computationally more efficient requiring 8 percent less CPU-time.

5. Conclusions

Two slightly different algorithms for the main part of the MOLP-procedure are presented in detail. One is essentially the commonly used ADBASE algorithm, and the other is a new variant, called MOLP-S, which needs 10-20% less computer-time when tested on a number of problems from the literature, varying in complexity. However, further research is needed and is currently in progress to compare both algorithms more fundamentally.

6. Areas for further research

A possible enhancement for both algorithms is the application of a procedure presented by Ecker and Kouada [4] which avoids checking the condition in theorem 1 for each pivotcolumn but uses all information contained in the actual tableau with a minimal number of Simplex-iterations. Note, however, that every improvement of a part, common in both MOLP-variants, would strengthen the advantage of the MOLP-S variant for the reported problems since in that case an equal quantity of work to be done by both variants would cost less time, rendering the relative difference in CPU-time in favour of MOLP-S.

APPENDIX

This appendix contains a detailed description of the two variations for mapping out the set of efficient extreme points. Let m denote the number of constraints and d the number of nonbasic variables. A Simplex-iteration is performed only on the right-hand-side and the nonbasic columns, extended with the "z-c"-part for the objective functions. The indices of the (non)basic variables are recorded in special arrays (j[1..d] for the nonbasic and i[1..m] for the basic variables. A pivot is defined by a row number r, identifying a basic variable i[r], and a column number c, corresponding with nonbasic variable j[c]. After the Simplex-iteration variable i[r] becomes nonbasic and its coefficients are stored in column c. The arrays i and j are adjusted accordingly. By keeping track of all pivots we can pivot back to each BFS which has been obtained before. Therefore we maintain an array P with (varying) length n for the recording of these pivotsteps. Let a BFS J be defined by the corresponding set of indices of nonbasic variables {j[1],...,j[d]}. Let J_0 be the efficient BFS found in step 2 of the MOLP-procedure. Furthermore, let x(J) be the extreme point corresponding with BFS J, and let z(J):=J∪{indices of basic variables with value 0}. In the following pseudo-code the begin-end's are omitted since the structure is obvious by the lay-out used. Furthermore, statements like P[n]:=(r,c) and S:=S {J} are not feasible in PASCAL, but are used to compress the pseudo-code. Some comments are added in italics on the right to clarify the variable names and the program code.

Step 3 of the MOLP-S variant.

```
PROCEDURE StepForward;
  (r,c):=AltPiv[n,NumAltPiv[n]]; From all existing alternative pivots one
  P[n] := (r,c);                    is taken and used for a step forward.
  NumAltPiv[n]:=NumAltPiv[n]-1;
  using pivot (r,c) leads to Bfs J;
  IF z(J)=J THEN
    list x(J); NumNdp:=NumNdp+1  The new efficient BFS is not degenerated.
  ELSE
    IF z(J)∉Z THEN  The new eff. BFS is degenerated but the corresponding
```

```
    Z:=Z∪{z(J)}; list x(J); NumNdp:=NumNdp+1;        extreme point is new.
PROCEDURE StoreAltPiv;
  S:=S∪{J};                       S contains all efficient BFS's found thus far.
  NumEffBfs:=NumEffBfs+1;                  Counting of number of efficient BFS's.
  NumAltPiv[n]:=NumAltPiv[n]+1;   Storage of alternative pivot leading to
  AltPiv[n,NumAltPiv[n]]:=(r,c);                                an efficient BFS.

n:=0; S:={J_0}; Z:={z(J_0)};
ready:=false; NumEffBfs:=1; NumNdp:=1;
REPEAT
  NewAdjEffBfs:=false;
  n:=n+1; NumAltPiv[n]:=0;
  FOR c:=1 to d DO
    determine all alternative pivots (r_1,c)..(r_a,c);
    IF a>0 THEN
      REPEAT
        r:=r_a; a:=a-1;
        NewBfs:=(pivot (r,c) would lead to Bfs J∉S);
      UNTIL NewBfs OR a=0;
      IF NewBfs THEN               A pivot is found which lead to a new BFS.
        IF (the condition in theor. 1 is satisfied for pivot col. c) THEN
          NewAdjEffBfs:=true;                        A new efficient BFS is found.
          StoreAltPiv;   The pivot leading to the efficient BFS is stored.
          WHILE a>0 DO   Each other alternative pivot with pivot column c
            r:=r_a; a:=a-1; and leading to an efficient BFS is stored when
            IF (pivot (r,c) would lead to BFS J∉S) THEN    this BFS is not
              StoreAltPiv;                                   registered before.
  IF NewAdjBfs THEN StepForward
  ELSE
    n:=n-1;
    IF n=0 THEN ready:=true
    ELSE
      backward:=true;
      REPEAT              Step backward until pathlength=0 or a step forward
                          can be made.
```

```
        (r,c):=P[n];
        use pivot (r,c);
        IF NumAltPiv[n]>0 THEN backward:=false; StepForward
        ELSE
          n:=n-1;
          IF n=0 THEN
            backward:=false; ready:=true;
      UNTIL NOT backward;
UNTIL ready;
```

The algorithm performs a reinversion when the number of Simplex-operations exceeds a certain prior chosen number. Note that a BFS J is added to S as soon as it is "adjacent efficient" according to theorem 1. The cardinality of S can grow to a very large number. Hashing techniques and linked lists are used to evaluate the expression ($J \notin S$) in little time (see section 2). Expression ($z(J) \notin Z$) is evaluated in a similar way.

<u>Step 3 of the ADBASE variant.</u>

```
PROCEDURE StoreUnscanned;
  S:=S∪{J};
  NumEffBfs:=NumEffBfs+1;
  NumUnscanned:=NumUnscanned+1;    Unscanned contains all eff. BFS's which
  Unscanned[NumUnscanned]:=J; are not yet scanned for new adj. eff. BFS's.

S:={J_0}; Z:={z(J_0)}; ready:=false; J_last:=J_0;
NumEffBfs:=1; NumUnscanned:=0; NumNdp:=1;
REPEAT
  FOR c:=1 to d DO
    determine all alternative pivots (r_1,c)..(r_a,c);
    IF a>0 THEN
      REPEAT
        r:=r_a; a:=a-1;
        NewBfs:=(pivot (r,c) would lead to Bfs J∉S);
      UNTIL NewBfs OR a=0;
```

```
      IF NewBfs THEN               A pivot is found which lead to a new BFS.
        IF (the condition in theor. 1 is satisfied for pivot col. c) THEN
          StoreUnscanned;           The BFS is efficient and will be stored.
          WHILE a>0 DO  Each other adjacent BFS which can be reached with
            r:=r_a; a:=a-1; pivot column c is eff. and will be stored when
            IF (pivot (r,c) would lead to BFS J∉S) THEN     the BFS is not
              StoreUnscanned;                             registered before.
    IF NumUnscanned>0 THEN
      J:=Unscanned[NumUnscanned];
      NumUnscanned:=NumUnscanned-1;
      pivot from J_last to J;
      IF z(J)=J THEN list x(J); NumNdp:=NumNdp+1
      ELSE
        IF z(J)∉Z THEN
          Z:=Z∪{z(J)}; list x(J); NumNdp:=NumNdp+1;
      J_last:=J;
    ELSE ready:=true;
  UNTIL ready;
```

Pivoting from J_{last} to J will often be performed in one Simplex-iteration since for each efficient basis all adjacent efficient bases are stored at the end of the array Unscanned. If more than one Simplex-iteration is necessary an artificial objective function with coefficients equal to 1 for the variables $J\backslash J_{last}$ and 0 for the other variables is minimized in order to pivot from J_{last} to J.

References

1. C.L. Hwang and A.S.M. Masud, *Multiple Objective Decision Making - Methods and Applications: a State of the Art Survey*. Lecture Notes in Economics and Mathematical Systems 164. Springer-Verlag, Berlin (1979).
2. J.P. Evans and R.E. Steuer, Generating efficient extreme points in linear multiple objective programming: two algorithms and computing experience, In *Multiple Criteria Decision Making (J.L. Cochrane and M. Zeleny, eds)*. University of South Carolina Press, Columbia, 349-365 (1973).
3. M. Zeleny, *Linear Multiobjective Programming*. Lecture Notes in Economics and Mathematical Systems 95, Springer-Verlag, Berlin (1974).
4. J.G. Ecker and I.A. Kouada, Finding all efficient extreme points for multiple objective linear programs. *Math. Prog.* **14**, 249-261 (1978).
5. R. Hartley, Survey of algorithms for vector optimization problems, In *Multi-Objective Decision Making (S. French, R. Hartley, L.C. Thomas and D.J. White, eds)*. Academic Press, London (1983).
6. H. Isermann, The relevance of duality in multiple objective linear programming, *TIMS Studies in the Management Sciences* **6**, 241-262 (1977).
7. T. Gal, A general method for determining the set off all efficient solutions to a linear vectormaximum problem. *Eur. J. Opl Res.* **1**, 307-322 (1977).
8. H. Isermann, The enumeration of all efficient solutions for a linear multiple-objective transportation problem. *Naval Res. Log. Q.* **26**, 123-139 (1979).
9. H. Isermann and R.E. Steuer, Computational experience concerning payoff tables and minimum criterion values over the efficient set. *Eur. J. Opl Res.* **33**, 91-97 (1987).
10. J.S.H. Kornbluth, Shadow prices in multiple objective linear programming. *J. Opl Res. Soc.* **36**, 1095-1101 (1985).
11. J. Philip, Short Communication: vector maximization at a degenerate vertex. *Math. Prog.* **13**, 357-359 (1977).

12. T.R. Rakes and G.R. Reeves, Selecting tolerances in chance-constrained programming: a multiple objective linear programming approach. *Operations Research Letters* **4**, 65-69 (1985).
13. R. Rhode and R. Weber, The range of the efficient frontier in multiple objective linear programming. *Math. Prog.* **28**, 84-95 (1984).
14. R.E. Steuer, *Multiple Criteria Optimization: Theory, Computation and Application.* Wiley, New York, (1986).
15. M. Zeleny, *Multiple Criteria Decision Making.* McGraw Hill, New York (1982).

IN 1989 REEDS VERSCHENEN

368 Ed Nijssen, Will Reijnders
"Macht als strategisch en tactisch marketinginstrument binnen de distributieketen"

369 Raymond Gradus
Optimal dynamic taxation with respect to firms

370 Theo Nijman
The optimal choice of controls and pre-experimental observations

371 Robert P. Gilles, Pieter H.M. Ruys
Relational constraints in coalition formation

372 F.A. van der Duyn Schouten, S.G. Vanneste
Analysis and computation of (n,N)-strategies for maintenance of a two-component system

373 Drs. R. Hamers, Drs. P. Verstappen
Het company ranking model: a means for evaluating the competition

374 Rommert J. Casimir
Infogame Final Report

375 Christian B. Mulder
Efficient and inefficient institutional arrangements between governments and trade unions; an explanation of high unemployment, corporatism and union bashing

376 Marno Verbeek
On the estimation of a fixed effects model with selective non-response

377 J. Engwerda
Admissible target paths in economic models

378 Jack P.C. Kleijnen and Nabil Adams
Pseudorandom number generation on supercomputers

379 J.P.C. Blanc
The power-series algorithm applied to the shortest-queue model

380 Prof. Dr. Robert Bannink
Management's information needs and the definition of costs, with special regard to the cost of interest

381 Bert Bettonvil
Sequential bifurcation: the design of a factor screening method

382 Bert Bettonvil
Sequential bifurcation for observations with random errors

383 Harold Houba and Hans Kremers
Correction of the material balance equation in dynamic input-output models

384 T.M. Doup, A.H. van den Elzen, A.J.J. Talman
Homotopy interpretation of price adjustment processes

385 Drs. R.T. Frambach, Prof. Dr. W.H.J. de Freytas
Technologische ontwikkeling en marketing. Een oriënterende beschouwing

386 A.L.P.M. Hendrikx, R.M.J. Heuts, L.G. Hoving
Comparison of automatic monitoring systems in automatic forecasting

387 Drs. J.G.L.M. Willems
Enkele opmerkingen over het inversificerend gedrag van multinationale ondernemingen

388 Jack P.C. Kleijnen and Ben Annink
Pseudorandom number generators revisited

389 Dr. G.W.J. Hendrikse
Speltheorie en strategisch management

390 Dr. A.W.A. Boot en Dr. M.F.C.M. Wijn
Liquiditeit, insolventie en vermogensstructuur

391 Antoon van den Elzen, Gerard van der Laan
Price adjustment in a two-country model

392 Martin F.C.M. Wijn, Emanuel J. Bijnen
Prediction of failure in industry
An analysis of income statements

393 Dr. S.C.W. Eijffinger and Drs. A.P.D. Gruijters
On the short term objectives of daily intervention by the Deutsche Bundesbank and the Federal Reserve System in the U.S. Dollar - Deutsche Mark exchange market

394 Dr. S.C.W. Eijffinger and Drs. A.P.D. Gruijters
On the effectiveness of daily interventions by the Deutsche Bundesbank and the Federal Reserve System in the U.S. Dollar - Deutsche Mark exchange market

395 A.E.M. Meijer and J.W.A. Vingerhoets
Structural adjustment and diversification in mineral exporting developing countries

396 R. Gradus
About Tobin's marginal and average q
A Note

397 Jacob C. Engwerda
On the existence of a positive definite solution of the matrix equation $X + A^T X^{-1} A = I$

398 Paul C. van Batenburg and J. Kriens
Bayesian discovery sampling: a simple model of Bayesian inference in auditing

399 Hans Kremers and Dolf Talman
Solving the nonlinear complementarity problem

400 Raymond Gradus
Optimal dynamic taxation, savings and investment

401 W.H. Haemers
Regular two-graphs and extensions of partial geometries

402 Jack P.C. Kleijnen, Ben Annink
Supercomputers, Monte Carlo simulation and regression analysis

403 Ruud T. Frambach, Ed J. Nijssen, William H.J. Freytas
Technologie, Strategisch management en marketing

404 Theo Nijman
A natural approach to optimal forecasting in case of preliminary observations

405 Harry Barkema
An empirical test of Holmström's principal-agent model that tax and signally hypotheses explicitly into account

406 Drs. W.J. van Braband
De begrotingsvoorbereiding bij het Rijk

407 Marco Wilke
Societal bargaining and stability

408 Willem van Groenendaal and Aart de Zeeuw
Control, coordination and conflict on international commodity markets

409 Prof. Dr. W. de Freytas, Drs. L. Arts
Tourism to Curacao: a new deal based on visitors' experiences

410 Drs. C.H. Veld
The use of the implied standard deviation as a predictor of future stock price variability: a review of empirical tests

411 Drs. J.C. Caanen en Dr. E.N. Kertzman
Inflatieneutrale belastingheffing van ondernemingen

412 Prof. Dr. B.B. van der Genugten
A weak law of large numbers for m-dependent random variables with unbounded m

413 R.M.J. Heuts, H.P. Seidel, W.J. Selen
A comparison of two lot sizing-sequencing heuristics for the process industry

IN 1990 REEDS VERSCHENEN

419 Bertrand Melenberg, Rob Alessie
A method to construct moments in the multi-good life cycle consumption model

420 J. Kriens
On the differentiability of the set of efficient (μ,σ^2) combinations in the Markowitz portfolio selection method

421 Steffen Jørgensen, Peter M. Kort
Optimal dynamic investment policies under concave-convex adjustment costs

422 J.P.C. Blanc
Cyclic polling systems: limited service versus Bernoulli schedules

423 M.H.C. Paardekooper
Parallel normreducing transformations for the algebraic eigenvalue problem

424 Hans Gremmen
On the political (ir)relevance of classical customs union theory

425 Ed Nijssen
Marketingstrategie in Machtsperspectief

426 Jack P.C. Kleijnen
Regression Metamodels for Simulation with Common Random Numbers: Comparison of Techniques

427 Harry H. Tigelaar
The correlation structure of stationary bilinear processes

428 Drs. C.H. Veld en Drs. A.H.F. Verboven
De waardering van aandelenwarrants en langlopende call-opties

429 Theo van de Klundert en Anton B. van Schaik
Liquidity Constraints and the Keynesian Corridor

430 Gert Nieuwenhuis
Central limit theorems for sequences with m(n)-dependent main part

431 Hans J. Gremmen
Macro-Economic Implications of Profit Optimizing Investment Behaviour

432 J.M. Schumacher
System-Theoretic Trends in Econometrics

433 Peter M. Kort, Paul M.J.J. van Loon, Mikulás Luptacik
Optimal Dynamic Environmental Policies of a Profit Maximizing Firm

434 Raymond Gradus
Optimal Dynamic Profit Taxation: The Derivation of Feedback Stackelberg Equilibria

435 Jack P.C. Kleijnen
Statistics and Deterministic Simulation Models: Why Not?

436 M.J.G. van Eijs, R.J.M. Heuts, J.P.C. Kleijnen
Analysis and comparison of two strategies for multi-item inventory systems with joint replenishment costs

437 Jan A. Weststrate
Waiting times in a two-queue model with exhaustive and Bernoulli service

438 Alfons Daems
Typologie van non-profit organisaties

439 Drs. C.H. Veld en Drs. J. Grazell
Motieven voor de uitgifte van converteerbare obligatieleningen en warrantobligatieleningen

440 Jack P.C. Kleijnen
Sensitivity analysis of simulation experiments: regression analysis and statistical design

441 C.H. Veld en A.H.F. Verboven
De waardering van conversierechten van Nederlandse converteerbare obligaties

442 Drs. C.H. Veld en Drs. P.J.W. Duffhues
Verslaggevingsaspecten van aandelenwarrants

443 Jack P.C. Kleijnen and Ben Annink
Vector computers, Monte Carlo simulation, and regression analysis: an introduction

444 Alfons Daems
"Non-market failures": Imperfecties in de budgetsector

445 J.P.C. Blanc
The power-series algorithm applied to cyclic polling systems

446 L.W.G. Strijbosch and R.M.J. Heuts
Modelling (s,Q) inventory systems: parametric versus non-parametric approximations for the lead time demand distribution

447 Jack P.C. Kleijnen
Supercomputers for Monte Carlo simulation: cross-validation versus Rao's test in multivariate regression

448 Jack P.C. Kleijnen, Greet van Ham and Jan Rotmans
Techniques for sensitivity analysis of simulation models: a case study of the CO_2 greenhouse effect

449 Harrie A.A. Verbon and Marijn J.M. Verhoeven
Decision-making on pension schemes: expectation-formation under demographic change